

Claims

- [c1] 1.A method for tracing the execution path of a computer program comprising at least one module including a plurality of instructions, at least one of said instructions being a branch instruction, the method comprising the steps of:
- identifying each branch instruction;
- evaluating each branch instruction to be one of true and false; and
- responsive to an evaluation of true, pushing a unique identifier into a predefined area of storage, wherein said unique identifier is associated with the instructions executed as a result of said evaluation of true.
- [c2] 2.The method of claim 1, wherein said predefined area of storage is in volatile memory.
- [c3] 3.The method of claim 1, wherein said predefined area of storage is in non-volatile memory.
- [c4] 4.The method of claim 1, comprising the step of:
- outputting the contents of said storage area to a file at a predetermined point in time.
- [c5] 5.The method of claim 4, comprising the step of:
- outputting trace information to said file upon exit from said at least one module.
- [c6] 6.The method of claim 5, wherein the contents of said storage area is outputted at the same time as said exit trace information.
- [c7] 7.The method of claim 4, wherein the step of outputting the contents of said storage area comprises:
- determining whether said storage area is full; and
- responsive to a positive determination, outputting said contents to said file.
- [c8] 8.The method of claim 4, wherein the step of outputting the contents of said storage area comprises:

determining whether a failure has occurred within said program; and
responsive to a positive determination, outputting said contents to said file.

[c9] 9.The method of claim 4, wherein the step of pushing a unique identifier into a predefined area of storage further comprises the steps of:
determining whether said predefined area of storage is full; and
overwriting the first unique identifier in said storage area.

[c10] 10.The method of claim 9, comprising the step of:
writing the position of the most recent unique identifier to be written out to said storage area to said storage area.

[c11] 11.The method of claim 10, wherein said position is used to determine the number of unique identifiers that have been overwritten prior to being written out to said file.

[c12] 12.The method of claim 11, comprising the step of:
responsive to determining that a large number of unique identifiers have been overwritten, increasing the size of said predefined area of storage.

[c13] 13.An apparatus for tracing the execution path of a computer program comprising at least one module including a plurality of instructions, at least one of said instructions being a branch instruction, said apparatus comprising:
means for identifying each branch instruction;
means for evaluating each branch instruction to be one of true and false; and
means, responsive to an evaluation of true, for pushing a unique identifier into a predefined area of storage, wherein said unique identifier is associated with the instructions executed as a result of said evaluation of true.

[c14] 14.A method for instrumenting a computer program comprising at least one module including a plurality of instructions, at least one of said instructions being a branch instruction, each branch instruction being evaluated to be one of true and false at run-time, with at least one signature instruction for indicating the execution path of said program at run-time, the method

comprising the steps of:

identifying each branch instruction;

identifying the instructions associated with an evaluation of true at run-time;

instrumenting said instructions associated with an evaluation of true with a signature instruction, wherein said signature instruction causes a unique identifier to be pushed into a predefined area of storage upon execution of said true instructions at run-time.

[c1 5]

15.A compiler for instrumenting a computer program comprising at least one module including a plurality of instructions, at least one of said instructions being a branch instruction, each branch instruction being evaluated to be one of true and false at run-time, with at least one signature instruction for indicating the execution path of said program at run-time, said compiler comprising:

means for identifying each branch instruction;

means for identifying the instructions associated with an evaluation of true at run-time;

means for instrumenting said instructions associated with an evaluation of true with a signature instruction, wherein said signature instruction causes a unique identifier to be pushed into a predefined area of storage upon execution of said true instructions at run-time.